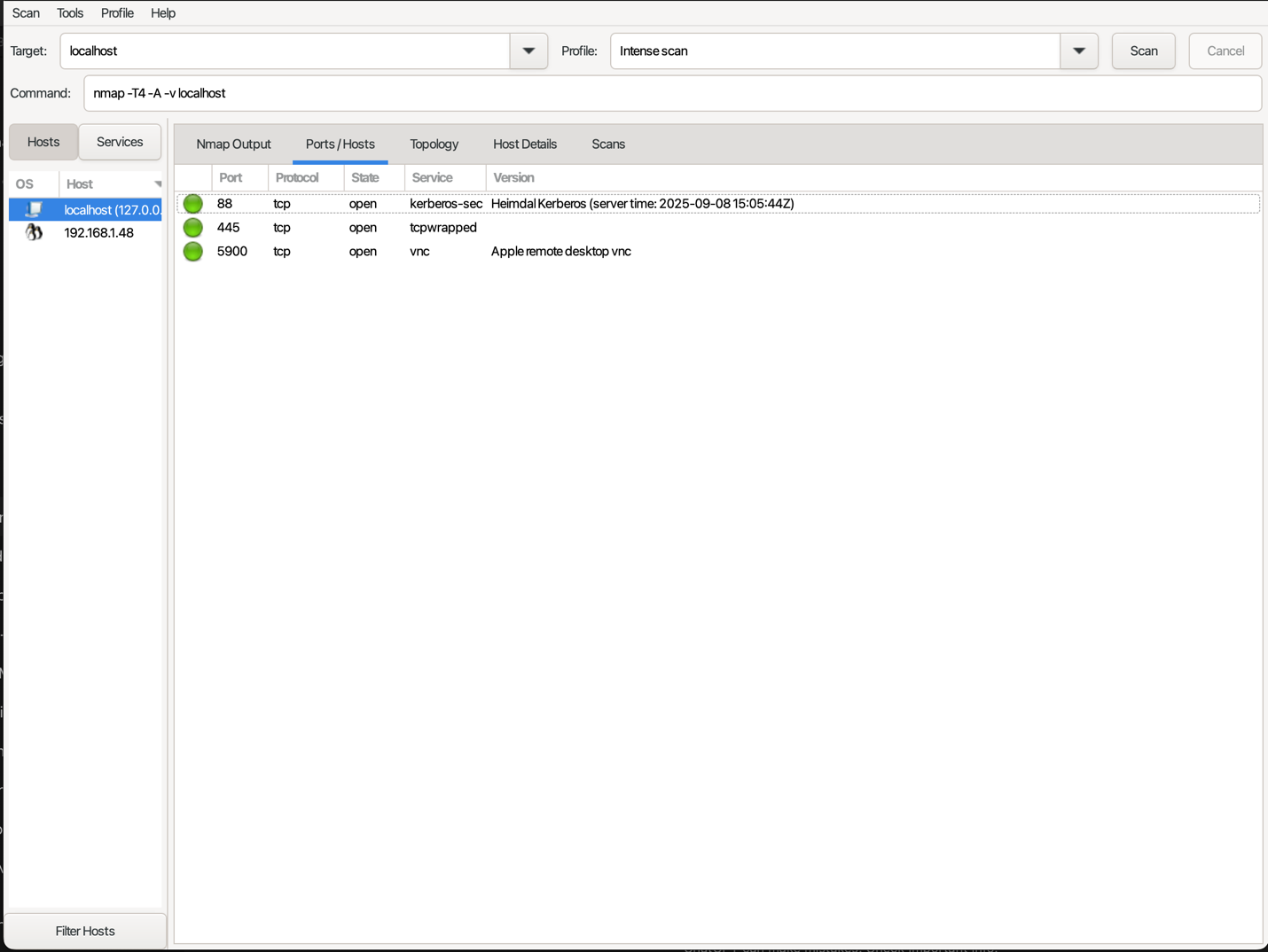
Activity 3

**Q1.**

Notice the open ports on all 3 devices (the attacker notebook, the target

notebook, and the target Linux VM). Does anything look suspicious, i.e., some ports

that you are not aware of that are open on the VM or on your notebooks?



**Attacker Notebook**

Looks suspicious at first, but no. Because 88(Kerberos), 445 (File Sharing), and 5900(VNC) are normal for Macbook.

A screenshot of a computer

AI-generated content may be incorrect.

**Target Notebook**

Not suspicious, 8800 is expected because the target notebook is running Uvicorn on port 8800.

A screenshot of a computer

AI-generated content may be incorrect.

**Target VM**

Not suspicious. Port 22 and 80 are SSH and TCP, which are normal.

**Q2.**

Look at the information provided by nmap about your OS's on all 3 devices. Is

the information correct? Why is it or why is it not correct?

Nmap’s OS information is not fully correct across the 3 devices because:

* OS fingerprinting relies on network stack quirks (TCP/IP responses), which can be very similar across different Unix-like systems.
* Firewalls and virtualization distort or block the probes Nmap uses.
* Service banners ≠ OS (just because it sees Kerberos/SMB/VNC doesn’t mean it knows it’s macOS).

So while the open port detection is accurate, the OS guesses are only approximate and may be misleading.

**Q3.**

What do you think about the information you can get using nmap? Scary?

Yes, it is actually scary. With just a single command, Nmap can show

* + open ports
  + running services
  + guess the operating system of a device

Now I understand the importance of closing unused ports and using strong firewalls.

**Q4.**

Look at the access.log file for the web server in your Linux VM. What IP

addresses do you see accessing the web server? Which devices do these IP addresses

belong to?

A screenshot of a computer program

AI-generated content may be incorrect.

192.168.1.42 Target Laptop Host IP

A screenshot of a computer program

AI-generated content may be incorrect.

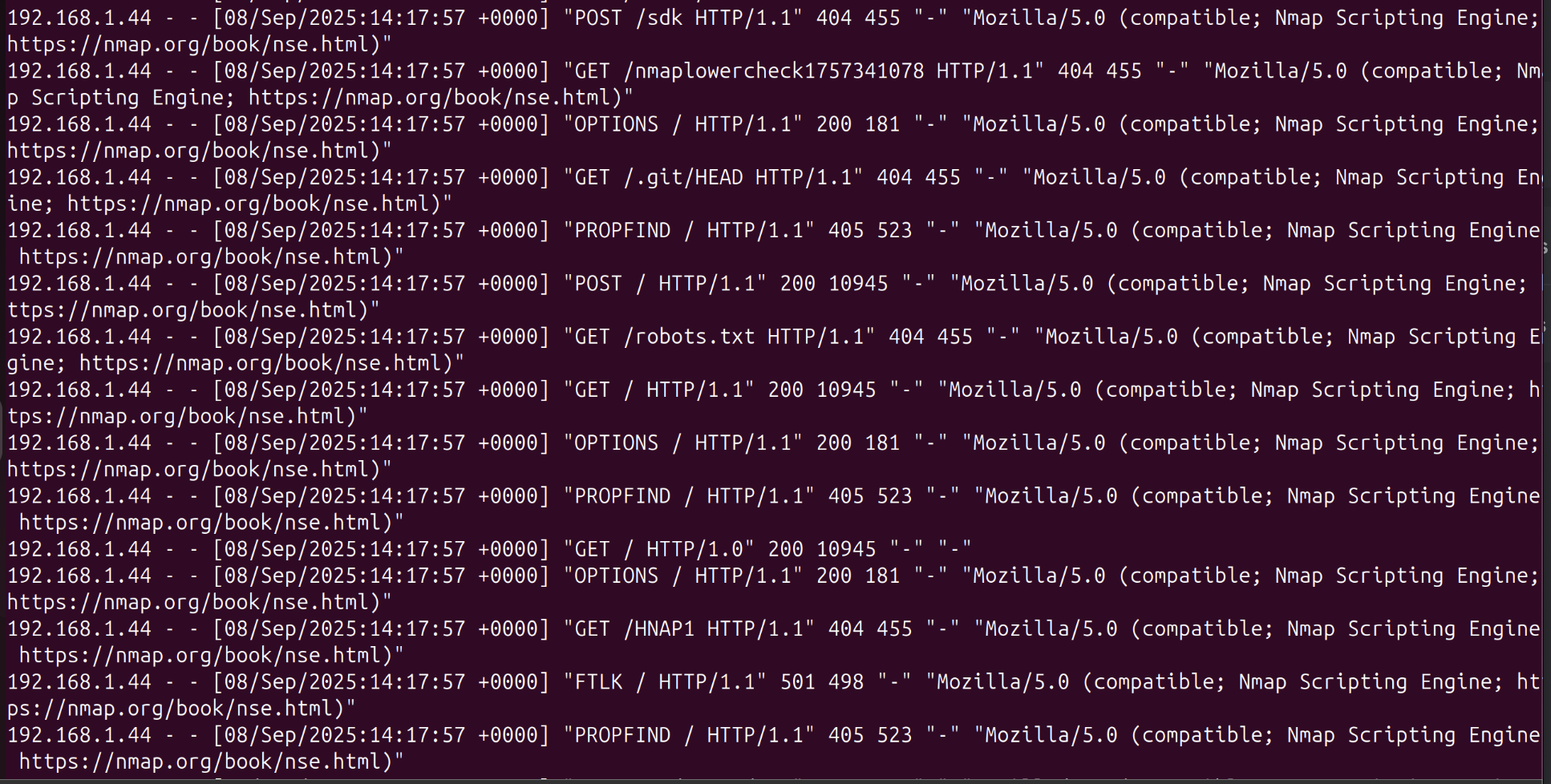
192.168.1.46 Attacker Laptop Host IP

192.168.1.150 Target VM Host IP

**Q5.**

Find the nmap scan in the web server log. Copy the lines from the log file that

were created because of the nmap scan.



A screenshot of a computer program

AI-generated content may be incorrect.

**Q6.**

After you successfully install your iptable rule(s), how do the reported results

from your new nmap scan compare to your previous scan before using iptables?

Look to see if OS detection, port open results, etc. have changed. Something(s) have

definitely changed.

A screenshot of a computer

AI-generated content may be incorrect.

**Before (1)**

A screenshot of a computer

AI-generated content may be incorrect.

**Before (2)**

A screenshot of a computer

AI-generated content may be incorrect.

**After (1)A screenshot of a computer

AI-generated content may be incorrect.**

**After (2)**

**A computer screen with white text

AI-generated content may be incorrect.**

**After (3)**

* NMAP only show PORT 80, PORT 22 missing.
* Closed port became filtered port.
* Can’t Ping.

**Q7.**

Notice that nmap can still figure out you have Apache httpd running. Look at

the access.log file for the web server in your Linux VM. Are the logs the same as in

Part II?

A screenshot of a computer program

AI-generated content may be incorrect.

**Log**

Yes, logs are the same.

**Q8.**

Explain whether or not you could prevent nmap from reaching the web server

while still allowing legitimate clients to get service. Will a firewall be suﬃcient for

this? Or do you need some other device? Please think critically about this.

**Can a firewall alone stop nmap but allow real clients?**

* **No.** A firewall can block unused ports, but if port 80/443 is open, nmap traffic looks like normal web traffic.
* Scanners can be stealthy (slow scans, fragmented packets, distributed sources).
* Firewall rules can’t reliably tell “scanner” from “browser.”

**What a firewall can do**

* Allow only required ports (deny everything else).
* Rate-limit connections and block obvious scan patterns.

**What’s needed beyond a firewall**

* VPN / client IP allowlisting → only trusted clients can connect.
* Mutual TLS (client certs) → server only accepts authenticated clients.
* Reverse proxy / CDN / WAF → hides the real server, filters malicious traffic.

**Q9.**

What are your firewall rules? Run iptables -L on your VM and enter the output

here.

Chain INPUT (policy DROP 2425 packets, 1209K bytes)

pkts bytes target prot opt in out source destination

9 1873 ACCEPT 0 -- lo \* 0.0.0.0/0 0.0.0.0/0

415 40409 ACCEPT 0 -- \* \* 0.0.0.0/0 0.0.0.0/0 ctstate RELATED,ESTABLISHED

67 3660 ACCEPT 6 -- \* \* 0.0.0.0/0 0.0.0.0/0 tcp dpt:80

0 0 ACCEPT 6 -- \* \* 192.168.1.42 0.0.0.0/0 tcp dpt:22

16 2608 DROP 1 -- \* \* 0.0.0.0/0 0.0.0.0/0

2 104 ACCEPT 6 -- \* \* 192.168.1.44 0.0.0.0/0 tcp dpt:22

Chain FORWARD (policy DROP 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 565 packets, 188K bytes)

pkts bytes target prot opt in out source destination